

# SYSTEM AND METHOD FOR CONNECTING BLUETOOTH-ENABLED DEVICES TO A PERSONAL COMPUTER

## BACKGROUND OF THE INVENTION

### Field of the Invention

[0001] The present invention relates to wireless networks. More specifically, the present invention relates to methods for connecting Bluetooth-enabled devices to other systems.

### Description of the Related Art

[0002] Computer networks allow multiple computers, peripherals and other information storage, retrieval or processing devices to share data. Each device attached to a network is typically referred to as a 'node'. Local Area Networks ("LANs") have historically consisted of nodes interconnected by physical media (e.g., coaxial cable, twisted pair wire, fiber optics, etc.). Recently wireless LANs, the nodes of which are not connected by means of a physical medium, have started to appear in the market. Wireless LANs communicate by means of infrared (IR), radio or other signals. One of the benefits of using wireless LANs is that cabling is not required. This is a particularly useful feature for mobile nodes such as laptop and notebook computers, PDAs (personal digital assistants), and the like. If equipped with an appropriate wireless adapter, the mobile nodes can move around within a predefined coverage area and remain connected to the network.

0004625 042701

[0003] One method of implementing a wireless LAN is similar to a cellular phone network system. In this method wireless mobile nodes do not communicate directly with each other, but rather send all signals to a central base station, which then redirects the signals to the destination node. In certain systems of this type, each wireless mobile node is allowed to simultaneously participate in different wireless networks.

[0004] A wireless technology called 'Bluetooth' is under development to enable ease of synchronization and mobility for a plethora of corporate and consumer applications. Specifications relating to Bluetooth are set forth in BLUETOOTH SPECIFICATION VERSION 1.0B CORE, published in December 1999, hereinafter referred to as the "Bluetooth standard". Bluetooth technology will open up many possibilities for communication via wireless networks. One potential application of Bluetooth-based technology involves the connection of peripheral devices to a Personal Computer (PC) or similar electronic device.

[0005] For an application on a PC to use a Bluetooth-enabled peripheral device, software must be present on the PC which is "Bluetooth-aware"; that is, which runs the Bluetooth protocol layers and is able to support a Bluetooth-based connection between the PC and the peripheral device. The conventional means of doing this is to provide software components called "device drivers" for the PC. Each potential Bluetooth-enabled peripheral then has a separate device driver installed on the PC to provide this functionality.

[0006] However, the difficulties with device drivers are well known. Device drivers are specific to the operating system and often to a particular version of an operating system. In addition, device drivers may be difficult to install and can interact undesirably with each other.

[0007] On the other hand, the recent advent of the Universal Serial Bus (USB) has spurred operating system manufacturers to provide similar drivers for common peripherals attached to a PC via a USB port. On Windows-based PCs, for example, these drivers are "plug and play"; Windows detects the attachment of a device to a

USB port and automatically installs the appropriate driver. Unfortunately, the plug and play capability of the Windows environment has heretofore not been extensible to peripheral devices over wireless links.

**[0008]** What is needed then, is a system or method for taking advantage of USB technology, and particularly the “plug and play” capability thereof, to simplify the installation and use of Bluetooth-enabled and other wireless peripherals.

### **SUMMARY OF THE INVENTION**

**[0009]** In summary, the present invention provides an interface for effecting communication between first and second systems. The interface includes a first arrangement for communicating with the first system using a first protocol and a second arrangement for communicating with the second system using a second protocol. At least one of the arrangements includes a mechanism for effecting wireless communication. In accordance with the present teachings, a third arrangement is included for translating data received from the first system in accordance with the first protocol into data adapted for the second system in accordance with the second protocol.

**[00010]** In the illustrative embodiment, the inventive interface further includes an arrangement for translating data received from the second system in accordance with the second protocol into data adapted for the first system in accordance with the first protocol. In the illustrative embodiment, the second system is a Personal Computer, the first system is a peripheral adapted to be connected to the computer, the first protocol is a Bluetooth protocol and the second protocol is a Universal Serial Bus (USB) protocol. In this context the inventive interface leverages USB technology, particularly “plug and play” capability, to simplify the installation and use of Bluetooth-enabled peripherals.

## BRIEF DESCRIPTION OF THE DRAWINGS

[00011] Fig. 1 is a block diagram of a system with a Bluetooth-enabled peripheral connected to a PC via a Bluetooth/USB interface of the present invention.

[00012] Fig. 2 is a block diagram of an illustrative embodiment of the Bluetooth/USB interface of the present invention.

[00013] Fig. 3 is a flow diagram showing an illustrative implementation of a method of operation of a Bluetooth/USB interface in accordance with the present teachings.

## DESCRIPTION OF THE INVENTION

[00014] Illustrative embodiments and exemplary applications will now be described with reference to the accompanying drawings to disclose the advantageous teachings of the present invention.

[00015] While the present invention is described herein with reference to illustrative embodiments for particular applications, it should be understood that the invention is not limited thereto. Those having ordinary skill in the art and access to the teachings provided herein will recognize additional modifications, applications, and embodiments within the scope thereof and additional fields in which the present invention would be of significant utility.

[00016] Fig. 1 is a block diagram of a system with a Bluetooth-enabled peripheral connected to a PC via a Bluetooth/USB interface of the present invention. The system 10 includes a PC 20 that is connected to a Bluetooth/USB interface (or "Dongle") 30 via a conventional USB interface 40. The Bluetooth/USB interface 30 communicates with a Bluetooth-enabled peripheral device 50 via antennas 60 and 70, respectively.

Those skilled in the art will appreciate that the present invention is not limited to the application shown nor the configuration thereof. That is, the Bluetooth/USB interface 30 may be connected to a device via a common interface, e.g., a USB interface, to effect communication with another device or system such as a PC via a Bluetooth link.

**[00017]** Fig. 2 is a block diagram of an illustrative embodiment of the Bluetooth/USB interface 30 of the present invention. The interface includes a Bluetooth device communication module 28, a USB communication module 29 and a USB-to-Bluetooth protocol translator 34 interposed therebetween. The Bluetooth device communication module 28 includes a peripheral detection and connection module 31, a Bluetooth multiplexer/demultiplexer 32 and a Bluetooth protocol stack 33. The USB communication module 29 includes a configuration database 36, a USB multiplexer/demultiplexer 37 and a USB protocol stack 38. The Bluetooth/USB interface 30 communicates through a conventional Bluetooth interface 39 and the USB interface 40. The interface 30 further includes a controller 35, which will typically comprise a microprocessor disposed to execute control software stored in associated memory (not shown).

**[00018]** Conventionally, when a peripheral device is plugged in to a USB port of a PC, the operating system engages in a standard dialog with the device to determine an appropriate driver. The present invention incorporates standard device emulation software into the Bluetooth/USB interface 30 allowing the interface 30 to "spoof" the attachment of one or more peripheral devices. The interface 30 then proceeds to translate between messages transferred over the Bluetooth wireless link and USB messages such that the PC 20 believes that the peripheral device 50 is wire-attached via the USB port rather than via the Bluetooth wireless link. This process is described in more detail below.

**[00019]** The peripheral detection and connection module 31 is connected to the Bluetooth interface 39 and operates under the control of the controller 35. The peripheral detection and connection module 31 may be implemented in software. The

peripheral detection and connection module 31 detects a connection of a peripheral device by receipt of a Bluetooth connection request message from the peripheral device 50 and a subsequent exchange of messages with the device 50 (to determine its type). Further, the peripheral detection and connection module 31 "handshakes" with the PC 20 to emulate the insertion of a device onto the USB bus of the device 50 (and emulates the removal of the device when necessary).

**[00020]** The Bluetooth protocol stack 33 is a software module which functions in accordance with the Bluetooth standard.

**[00021]** As discussed more fully below, the controller 35 is configured to respond to inquiries from, and establish connections with, appropriate Bluetooth-enabled peripheral devices. In operation, the controller 35 monitors the peripheral detection and connection module 31 for the arrival or departure of Bluetooth-enabled peripherals. When a Bluetooth-enabled peripheral is detected, the controller 35 establishes a Bluetooth connection and determines the type of emulation required for the newly connected device using a table lookup or other suitable technique. Next, the controller 35 informs the PC 20 of the "insertion" of a newly connected device. The controller 35 then switches in the appropriate protocol translation module 34 via the Bluetooth multiplexer 32 and the USB multiplexer 37. A set of protocol translation modules 34 is provided, one for each type of device to be supported. These modules enable the controller 35 to relays messages between the PC 20 and the device 50 via the Bluetooth protocol stack 33 and the USB protocol stack 38, translating as necessary between the ports to make the Bluetooth-attached peripheral 50 emulate a USB-attached peripheral. Each of the translators 34 may be implemented with a simple state machine or using other means well within the capability of one of ordinary skill in the art. The USB communication protocol stack 38 should include "plug and play" support.

**[00022]** When a device fails to respond (possibly due to moving out of range) the controller 35 announces to the PC 20 that the device has been "removed".

[00023] Fig. 3 is a flow diagram showing an illustrative implementation of a method of operation of a Bluetooth/USB interface implemented in accordance with the present teachings. In accordance with the illustrated method 100, at step 102, the system 30 waits in idle mode for an event to occur. At step 104, a peripheral device is detected by the peripheral detection and connection module 31. At step 106, the peripheral detection and connection module 31 directs the Bluetooth protocol stack 33 to open a connection. At step 108, the Bluetooth protocol stack 33 waits for a connection to be established. If no connection is established before a predetermined time period (i.e., the device 'times-out'), then the interface returns to step 102 to wait for another event. If, however, a connection is established at step 108, then, at step 110, the Bluetooth protocol stack 33 notifies the controller 35 of the connection and the control software 35 determines the device type. At step 112, the control software 35 selects the correct protocol translation module 34 and tells the Bluetooth multiplexer 32 of the new device 50. At step 114, based on configuration data stored in the configuration database 36, the translation module 34 generates device insertion messages to the USB protocol stack 38 via the USB multiplexer 37. At step 116, the USB protocol stack 38 handshakes with the PC 20 via the USB interface 40 and the interface 30 returns to step 102 to wait for another event.

[00024] In the event that data is received from the peripheral device 50 via the Bluetooth interface 39 (step 118) then, at step 120, the Bluetooth protocol stack 33 relays the data via the Bluetooth multiplexer 32 and the appropriate USB-to-Bluetooth protocol translation module 34. At step 122, the appropriate protocol translation module translates the data as necessary. Next, at step 124, the translation module 34 sends the data to the USB protocol stack 38 via the USB multiplexer 37. Finally, at step 126, the USB protocol stack 38 sends data to the PC 20 via the USB interface 40 and the interface 30 returns to step 102 to wait for another event.

[00025] If, at step 128, data is received from the PC 20 via the USB interface 40, then at step 130, the USB protocol stack 38 relays data to an appropriate translation module 34 via the USB multiplexer 37. Next, at step 132, the selected translation

module 34 translates the data as necessary. At step 134, the translation module 34 sends data via the Bluetooth demultiplexer 32 to the Bluetooth protocol stack 33. Finally, at step 136, the Bluetooth protocol stack 33 sends the translated data to the device 50 via the Bluetooth interface 39 and the interface 30 returns to step 102 to wait for another event.

**[00026]** If, at step 138, the connection to the peripheral device is lost, then at step 140, a 'connection lost' message is passed from the Bluetooth protocol stack to the protocol translation module 34 via the Bluetooth multiplexer 32. Next, at step 142, the protocol translation module 34 generates USB event code to emulate a device being unplugged and sends it to the USB protocol stack 38 via the USB multiplexer/demultiplexer 37. Finally, at step 144, the USB protocol stack notifies the PC 20 via the USB interface 40 that the device 50 is disconnected and the interface 30 returns to step 102 to wait for another event.

**[00027]** Thus, by incorporating the present invention in a Bluetooth Dongle, Bluetooth-enabled peripheral devices, which have a standard USB-attached analog, may be logically connected to any PC without installing custom software on the PC and without any of the drawbacks of custom drivers. Conventionally, as new versions of operating systems are produced, drivers for standard USB peripherals such as modems, printers, mice, etc. are updated if necessary, and tested by the OS manufacturer for interoperability. However, peripherals attached to a PC via an interface implemented in accordance with the present teachings may continue to operate without requiring custom driver development.

**[00028]** Thus, the present invention has been described herein with reference to a particular embodiment for a particular application. Those having ordinary skill in the art and access to the present teachings will recognize additional modifications, applications and embodiments within the scope thereof. For example, the invention may be configured to support multiple peripheral devices simultaneously connected via a simple Bluetooth radio and a single USB port. The extension to this case is straightforward to one skilled in the art and only requires configuring the controller 35



to keep track of multiple Bluetooth connections and direct the data through the appropriate translation module 34. Both the Bluetooth and USB protocols contain support for simultaneous connection of multiple devices.

[00029] It is therefore intended by the appended claims to cover any and all such applications, modifications and embodiments within the scope of the present invention.

Accordingly,

WHAT IS CLAIMED IS:

10000000.000000